

# Skeleton-guided 3D Shape Distance Field Metamorphosis

Bo Wu

HPCL, National University of Defense Technology

Shenzhen VisuCA Key Lab / SIAT

wubo.gfk@gmail.com

Kai Xu

HPCL, National University of Defense Technology

Shenzhen VisuCA Key Lab / SIAT

kevin.kai.xu@gmail.com

Yueshan Xiong

HPCL, National University of Defense Technology

ysxiong@nudt.edu.cn

Yang Zhou

Shenzhen VisuCA Key Lab / SIAT

zhouyang@siat.ac.cn

Hui Huang

Shenzhen University

Shenzhen VisuCA Key Lab / SIAT

hhzhiyan@gmail.com

## Abstract

We introduce an automatic 3D shape morphing method without the need of manually placed anchor correspondence points. Given a source and a target shape, our approach extracts their skeletons and computes the meaningful anchor points based on their skeleton node correspondences. Based on the anchors, dense correspondences between the interior of source and target shape can be established using earth movers distance (EMD) optimization. Skeleton node correspondence, estimated with a voting-based method, leads to part correspondence which can be used to confine the dense correspondence within matched part pairs, thus providing smooth and plausible morphing results based on distance field interpolation (DFI). We demonstrate our algorithm works well with experimental results, including shapes with large geometry variation and structure difference.

## 1. Introduction

Shape morphing (shape interpolation) [1, 11, 15, 27, 31, 32, 33] is a widely studied problem in both fields of computer vision and computer graphics. Smooth transition between 2D objects or even 3D shapes is very useful in many applications, such as special effects in film industry. For the same object in two different poses, shape morphing can give us the sequence of motions that move from one pose to the other. Interpolation between two different shapes with similar pose, can provide us with more shape appearance

variations. In general, these two cases happen simultaneously during shape blending.

In computer graphics, shape morphing approaches can be classified according to the representation of input objects. Surface mesh morphing methods [1, 33] can smoothly deform one mesh into another. While volumetric (*e.g.* distance field method) methods [11, 32] represent object as a grid and each node on the grid stores distance to the surface. By interpolating the distance values between source and target, it can generate grids with in-between distance values. At the cost of expensive computation, the distance field interpolation (DFI) method can handle morphing objects with different topologies, which is not achievable by explicit surface methods. Implicit surface methods [14, 20, 22, 26] defined by continuous functions are also well suited for morphing shapes of arbitrary topologies, but the components need to be bijectively paired.

The core issue of shape morphing is to establish meaningful correspondences between given objects. Surface morphing approaches usually need one-to-one correspondence between source and target meshes. DFI methods do not require dense correspondences at the beginning, however, it usually needs user to specify anchor points (usually more than a dozen on each object manually [11, 32]. In order to generate satisfactory in-betweens, these anchor points should be carefully placed in semantically meaningful positions of both source and target objects. It is a tedious work for human, and often hard to decide where to place anchors to achieve expected transition sequence.

In this paper, we propose an automatic DFI morphing method in part wise with the assistance of shape skeleton.

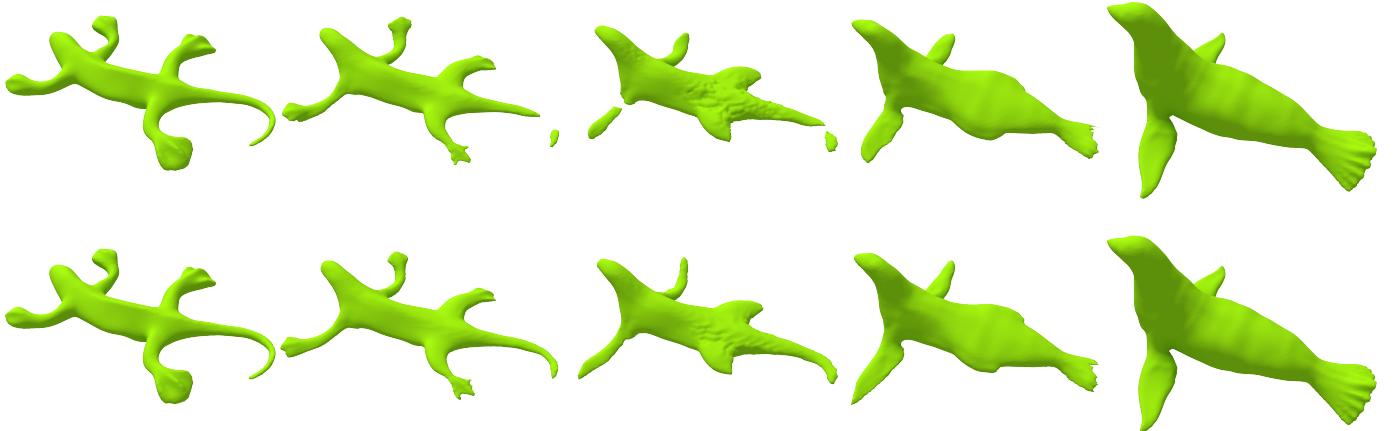


Figure 1: Morphing two different structural shapes *automatically*. Top row: morphing results of [32]. Bottom row: morphing sequence based on our algorithm. Note both results use the same anchors obtained by our method and there is no anchor on gecko’s two lower legs.

The correspondences of source and target skeleton feature nodes (*i.e.* junction and terminal nodes) are first established using a method similar to [5]. Based on the match of skeleton feature nodes, skeleton branch correspondences can also be derived. By parameterizing matched skeleton branches, more corresponding node pairs can be added as anchors. Briefly, skeleton is exploited in two aspects. First, it helps automatically determine where to place anchor points according to corresponding nodes on source and target skeletons. Second, during skeleton construction [38], shapes can be segmented into meaningful parts which are associated with skeleton branches. This association facilitates building dense correspondences in part wise via earth mover’s distance (EMD) optimization [19]. In this way, our algorithm can produce satisfactory morphing sequences without tedious user interaction.

The key observation of this paper is that skeleton of object, encoding both geometric and structural information, could supply us with meaningful positions for anchor points by parameterizing skeleton branches. Besides, part-based dense correspondences, instead of the overall correspondences between objects [32], avoid artifacts caused by correspondence drifts from EMD optimization. In this paper, we focus on generating smooth and plausible transitional shapes between source and target objects. It distinguishes ours from recent morphing work [2] also using skeleton, where they focus on synthesizing new creative in-betweens via blending part by part instead of overall transition.

## 2. Related work

Shape morphing is closely related to shape correspondence problem. Extensive literature exists on both shape morphing and shape correspondence topics. In this sec-

tion, we briefly review related shape morphing methods and shape matching on skeleton.

**Shape Morphing.** Since shape morphing is a useful tool in many applications, researches on both 2D and 3D morphing start quite early [7, 11, 21, 22]. Morphing methods can be classified into several kinds, *e.g.* surface mesh method, implicit surface method and DFI etc., according to how objects are represented. However, such methods rely heavily on the quality of correspondences to produce plausible morphing sequences. Surface mesh morphing approach [1, 33] needs dense correspondences for each vertex on source and target objects, while DFI [11, 32] needs correspondences of voxels on grids. Without bothering with voxels, implicit surface methods [20, 27] defined by scalar functions, can also morph objects with different topologies. In order to save effort on vertex correspondence in surface based morphing, Blanz and Vetter [9] proposed a morphable model with full correspondence to synthesize new faces from given examples. Similar work [3] has also been done for human poses. But such model is not applicable to DFI method, as it is implicit. In practice, given only sparse correspondences, some strategies can be used to generate fuzzy dense correspondence to make DFI work [11, 32].

Many works have been done to find a reasonable trajectory for morphing. The well known one is done by Alexa *et al.* [1], in which they proposed a new way to decompose the deformation gradient matrix into a rigid part and a stretching part. By interpolating each part separately to get a new deformation matrix, the blending path is visually reasonable. This method, however, was designed for surface or volumetric mesh, and is not straightforward for DFI. Weng [32] introduced such method into DFI by essentially constructing a volume mesh from grid, and their

results are promising. Besides, Xu *et al.* [33] introduced a method to obtain shape blending sequences by solving Poisson equation. But the deformation matrix decomposition is similar to [1]. There are also some works [12, 16, 17, 28] embedded physical model constraint into the morphing process, in order to produce physically plausible transformation sequences. Another recent work [15] proposed a data-driven method to yield reasonable morphing results. More recently, Von-Tycowicz *et al.* [31] exploited a set of shapes to generate real-time non-linear shape interpolation. Obviously, such methods depend on a dataset, in which all models are fully corresponding to each other. On the contrary, this also limits the method for arbitrary shapes.

**Skeleton matching.** Shape blending results greatly depend on the quality of correspondence. Establishing meaningful shape correspondence is a difficult problem, especially semantically similar objects may vary significantly in both geometry and topology. Shape matching is extensively studied and comprehensive review of this topic is beyond the scope of this paper. Please refer to survey papers like [30] for more details. In this paper, we focus on the correspondence of shape skeleton.

Curve skeleton contains both geometry and structure information about the shape. But how to efficiently extract skeleton from arbitrary shape is nontrivial. Several methods [4, 29] have been proposed to automatically extract high quality shape skeleton. Even recently, Zhou *et al.* [38] proposed a method to decompose shape into approximate generalized cylinders and one of its applications is to construct skeleton. In this paper, we employ this method to construct skeleton as it also gives us part information. Skeleton is also very useful in many aspects. Zheng *et al.* [37] made use of consensus skeleton to register point cloud with noise and occlusion. Jiang *et al.* [18] exploited skeleton to detect intrinsic symmetry of point clouds.

Establishing skeleton correspondences is a challenging task. Bai and Latecki [6] proposed a geodesic distance based skeleton matching algorithm, in which they only considered the match of skeleton endpoints. To avoid extensive computation, Au *et al.* [5] introduced a voting strategy which used several geometric metrics to search skeleton node correspondences. However, few works have been done to exploit skeleton for morphing. Blanding *et al.* [8] simply regarded the medial axis as the intermediate shapes. Lian and Xiao [23] made use of skeleton, strokes and key points to interpolate the same Chinese character in different fonts. Alhashim *et al.* [2] also exploited skeleton correspondence to generate creative shapes. We prefer using skeleton correspondence not only because it is computationally more efficient than vertex correspondence [36], but skeleton nodes often locate in the meaningful positions of a shape.

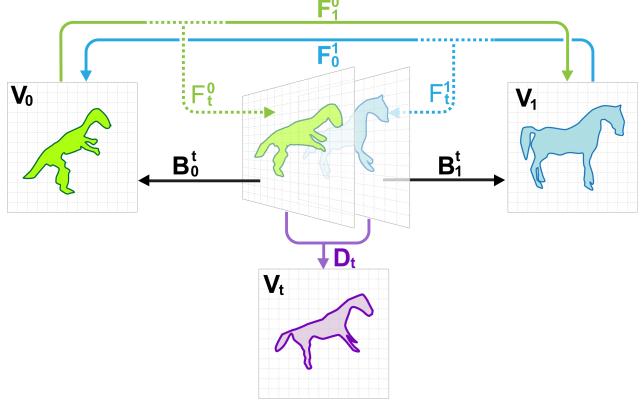


Figure 2: Framework of DFI based morphing method.

### 3. DFI morphing

Given a source and a target surface meshes, their occupied space is first uniformly voxelized as volume objects  $V_0$  and  $V_1$  respectively. For  $i^{\text{th}}$  voxel  $v_i^0 \in V_0$  ( $v_i^1 \in V_1$ ), it has a unique coordinate in source (target) domain and  $D_0(v_i)$  ( $D_1(v_i)$ ) is its distance to surface boundary, where positive value means outside, and negative means inside. Suppose there exists a third uniformly voxelized space domain  $V_t$  at time  $t$  ( $t \in [0, 1]$ ), DFI method constructs the intermediate shape by assigning distance values to voxels in  $V_t$ . DFI method is usually composed of warping and interpolation steps to construct better intermediate shapes [11].

The warping step can be divided into *forward* warping and *backward* warping. Given a bunch of voxel correspondences, a forward warping function can be computed to transform source domain  $V_0$  or target domain  $V_1$  to the intermediate voxel domain  $V_t$ . The backward warping function, on the contrary, deforms intermediate voxel domain to source domain or target domain. Specifically, the forward warping function from source domain to target domain (or from target to source) is first computed according to the given voxel correspondences between  $V_0$  and  $V_1$ . Applying the forward warping function to all source (target) domain, each voxel in source (target) will have a corresponding positions in target (source):

$$F_t^0(V_0) \approx V_1, \quad F_t^1(V_1) \approx V_0 \quad (1)$$

Note that  $F_t^0$  is unnecessary same as the inverse of  $F_0^1$ , and vice versa. With  $F_t^0$  and  $F_t^1$  in hand, different interpolation schemes could be used to obtain the corresponding intermediate voxels  $V_t^0$  and  $V_t^1$  separately. Starting from here, the forward warping functions  $F_t^0$  and  $F_t^1$  can be applied to update their correspondences in intermediate voxel domain  $V_t$ , and then its backward warping functions  $B_t^0$  and  $B_t^1$  can also be obtained in a similar way. Finally applying

the backward warping functions, the correspondences of all  $V_t$  voxels in source domain and target domain are achieved:

$$B_0^t(V_t) \approx V_0, \quad B_1^t(V_t) \approx V_1 \quad (2)$$

Here we use as-rigid-as-possible (ARAP) interpolation method in [32] to calculate in-between voxel positions corresponding to source and target respectively. As for warping function, radial basis function (RBF) [10] with a global affine transform is exploited, and thin plate spline is chosen as the kernel function of RBF. Now the distance value  $D_t$  for each voxel in  $V_t$  can be linearly interpolated from source domain and target domain:

$$D_t(V_t) = (1 - t)D_0(B_0^t(V_t)) + tD_1(B_1^t(V_t)) \quad (3)$$

The whole framework of DFI morphing is shown in Figure 2. In order to visualize intermediate shapes, isosurface of  $V_t$  is extracted using marching cubes [24] for rendering. As we see, correspondence is the key to compute warping functions, and also is very important for morphing. In section 4, we will explain how anchor points are automatically placed to compute dense correspondences and improve the quality of correspondence in voxel level.

#### 4. Skeleton-guided DFI morphing

Traditional DFI methods [11, 32] require user to place anchor points manually to interpolate two shapes smoothly. The number of corresponding anchor points is usually more than a dozen on both source and target objects. To achieve satisfying morphing sequence, these anchor points should be carefully placed in meaningful positions. It is a challenging and tedious work even for people in computer graphics field, not to mention those who know little about geometry.

In order to simplify this procedure, we introduce a morphing algorithm that automatically places anchor points with the aid of skeleton. The key observation is that shape skeleton conveys predominant geometry and structure information of the shape, and its nodes are often located in the meaningful positions inside the shape. Based on skeleton node correspondences, we can automatically determine corresponding anchors on source and target. Then EMD optimization is done in part wise to avoid large correspondence drifts. To obtain visually plausible in-betweens, source and target objects are assumed to have the same orientation. We will detail our method in following sections.

##### 4.1. Skeleton construction

The first step of our algorithm is to extract skeletons from input shapes. Here the skeleton is defined as rotational symmetry axis of surface meshes or point clouds and constructed using a recent algorithm proposed by Zhou *et al.* [38]. This method first over segments mesh into a large mount of small parts that are approximate to generalized



Figure 3: Shape decomposition and skeleton. Left column: shape decomposition and its skeleton branches. Right column: skeleton is connected as interpolated one.

cylinder. Then the local cylinders are merged into a dozen of long generalized cylinders. Finally, all the possible combinations of long cylinders are enumerated and the one with lowest cost is chosen. Skeleton is naturally obtained during the process, as each generalized cylinder intrinsically has a rotational symmetry axis as a skeleton branch. The chosen combination connects corresponding skeleton branches as an integrated skeleton.

The advantage of such skeleton is that it associates each skeleton branch with a shape segment. This helps us realize establishing dense correspondences for each matched part pairs later. Be aware that the decomposition and skeleton construction do not need any user intervention. Figure 3 show results of shape decompositions, skeleton branches and integrated skeletons. In the following, the hand model and plant model are used to illustrate our algorithm.

##### 4.2. Automatic anchor placement

Determining anchor points between source and target shapes is essentially a shape correspondence problem. It is widely regarded as a difficult problem. Directly establishing vertex correspondences between source and target meshes, *e.g.* via deformation-driven method [36], is very costly. A skeleton branch matching method is proposed in [38] to find one-to-one branch correspondences. But their method may get in trouble, when source and target skeletons have different number of branches. Therefore, we resort to a fast and accurate voting-based method [5] to compute correspondences of skeleton nodes, which could also handle partial matching problem. Here we only consider terminal nodes ( $degree = 1$ ) and junction nodes ( $degree > 2$ ), as terminal nodes are the extremities of shape and junction nodes con-

vey shape topology information. The correspondences of feature nodes between source and target skeletons are voted using metrics [5], such as centricity, path length, topology consistency, spatial configuration, etc. Since source and target objects have the same orientation, they can be roughly aligned just by translation. Therefore, we extend the method with considering the euclidean distance metric to alleviate symmetry-switching problem.

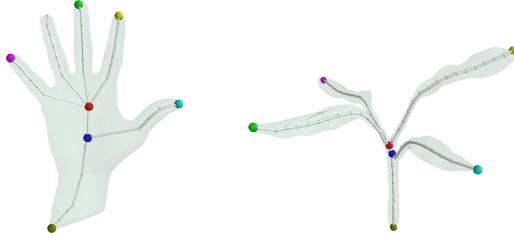


Figure 4: Match of feature nodes. The corresponding node pairs are shown in the same colors. Note that the terminal node of ring finger is left unmatched.

Figure 4 illustrates the match of feature nodes. Since feature node correspondences are one-to-one, not all feature nodes of a skeleton can have a corresponding node on another skeleton. For instance, no feature node of plant skeleton is matched with the terminal node of ring finger.

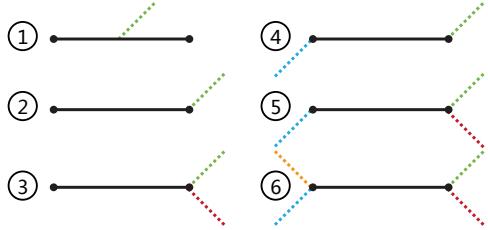


Figure 5: All six different cases of connecting one branch (solid line) into an integrated skeleton. In each case, dotted lines in different colors indicate other skeleton branches.

As feature node correspondences are established already, these nodes can be directly used as anchors. However, the number of such anchors is usually too few to obtain satisfactory morphing results, as they cannot supply enough information to attain good in-betweens. According to the way that skeletons are constructed [38], we propose a simple method to establish skeleton branch correspondences based on feature node correspondences. For one skeleton branch being merged into the integrated skeleton, one of six different cases shown in Figure 5 happens. If at least one of branch's two endpoints is treated as terminal node (case 1, 2, 3), terminal node matching signifies the match of two branches containing the terminal nodes. If a branch

with two terminal nodes matches two branch, we randomly choose one as its corresponding branch. Branches with no feature node (case 4) or one junction node (case 5) are left unmatched. If branch has two junction nodes (case 6), it will match another branch with both feature nodes matched. Such simple strategy works well in our experiments, because the matched feature nodes are guaranteed to have similar spatial configuration. The situation like case 4 and 5 happens seldom and has little effect on morphing results.

Since we have acquired the correspondences of original skeleton branches, each pair of corresponding branches is parametrized into  $[0, 1]$  using cubic B-spline. Points with same parametrized values are regarded as correspondences. In this way, more anchor points can be simply added for morphing. In all our experiments, we uniformly sample points on each branch as potential anchor points. Figure 6 illustrates locations of all the corresponding anchor points.

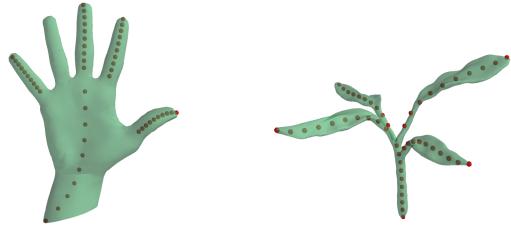


Figure 6: Corresponding anchor points. Note that there is no anchor point placed on the ring finger.

#### 4.3. Part-based EMD optimization

To achieve satisfactory blending results, dense correspondences are required to control locations of interior voxels during morphing. Inspired by Weng *et al.* [32], we also make use of EMD optimization to generate dense correspondences in voxel level from anchor points.

Harmonic fields [34, 35] are first computed using anchor points obtained from section 4.2. Different from [32], our harmonic fields are computed only on interior voxels of source and target. It is essentially a Laplace equation with a set of Dirichlet boundary conditions. Its discrete formulation on the  $i^{\text{th}}$  anchor point is:

$$h_i(v) - \frac{1}{n} \sum_{u \in N(v)} h_i(u) = 0, \text{ with} \\ h_i(p_i) = 1, h_i(p_j) = 0, j \neq i \quad (4)$$

In equation 4,  $N(v)$  is facet adjacent voxels of  $v$ ,  $n$  is the size of  $N(v)$ ,  $p_i$  ( $p_j$ ) is the nearest interior voxel of  $i^{\text{th}}$  ( $j^{\text{th}}$ ) anchor point. This equation can be equivalently solved via quadratic energy minimization. Solving such equation on each anchor point, we obtain a  $K$ -dimensional ( $K$  is the number of anchors) vector field  $\mathbf{h}^0$  ( $\mathbf{h}^1$ ) on source (target)

interior voxels. Figure 7 shows one harmonic field result on the hand model. It can also be observed that the harmonic values on voxels decrease quickly, which is another reason that more anchors are needed to distinguish themselves to achieve better correspondences.

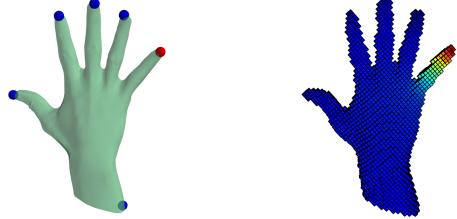


Figure 7: Harmonic field on voxelized model. The left shows hand model with Dirichlet boundary setting (blue nodes 0 and red node 1). The right shows the resulting harmonic field.

Using the vector field to measure the similarity of source and target voxels, EMD problem can be formulated to minimize the following transport cost:

$$\begin{aligned} \operatorname{argmin}_f \quad & \sum_{i,j} \left\| \mathbf{h}^0(v_i^0) - \mathbf{h}^1(v_j^1) \right\| \cdot f(i,j) \\ \text{s.t.} \quad & f(i,j) \geq 0 \\ & \sum_j f(i,j) = 1/N_0 \\ & \sum_i f(i,j) = 1/N_1 \end{aligned} \quad (5)$$

Since both source and target are uniformly voxelized, we assume each voxel in source or target has the same mass, which implies  $N_0$  and  $N_1$  are constant numbers in equation 5. This problem can be solved using the network simplex algorithm [13] and the optimized object function  $f$  tells us how much mass is transported from any interior voxel  $v_i^0$  of source to any interior voxel  $v_j^1$  of target. Therefore, the function can be used as weights to acquire a corresponding position in target (source) domain for each interior voxel in source (target).

In [32], both source and target objects are assumed to contain the same mass as 1, which means  $N_0$  and  $N_1$  in equation 5 are the numbers of source and target interior voxels respectively. This assumption works well if semantically corresponding parts in source and target have similar proportions. However, in case corresponding object parts have highly different proportions, the optimized function will provide us unexpected correspondences. Consequently, it will cause large shape distortion during morphing.

We instead introduce a part-based EMD optimization to compute the dense correspondences. Since the correspondences between source and target skeleton branches have

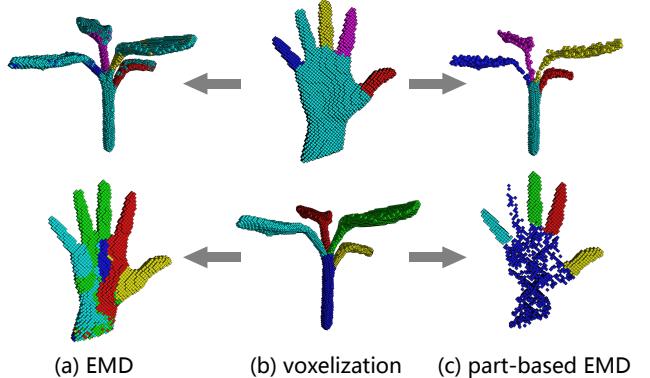


Figure 8: Comparison of EMD optimization and part-based EMD optimization.

been obtained in section 4.2, the voxels associated with corresponding skeleton segments are regarded to be matched as well. Therefore, we employ EMD optimization 5 on each pair of corresponding parts (now  $N_0$  and  $N_1$  are the voxel numbers inside parts) to achieve high quality correspondence. Figure 8 demonstrates exactly the case that corresponding parts have large different proportions. Voxelized source and target objects are shown in the middle column. For convenience, different parts are shown in different colors. The parts in source (target) which are not matched in target (source), e.g. ring finger of the hand, are merged into it connecting matched part, e.g. the ring finger is merged into the palm.

From top row of Figure 8, it can be observed that voxels inside the palm have their correspondences distributed in most regions of the plant, although the palm should mainly correspond to stem of the plant from view of structure. It is just because the palm occupies a large amount of mass, while its corresponding part of the plant, i.e. the stem, contains only a very small amount. In such case, the result of EMD optimization will assign voxels from other regions of the plant as correspondences of the voxels in palm. Such correspondences will lead to a undesired warping function and eventually cause large shape distortion during morphing. It is nontrivial to filter such correspondences out to solve the problem completely. In the bottom row of Figure 8, it shows a similar situation from target (plant) to source (hand). While the last column in Figure 8 shows the voxel correspondences using our part-based EMD optimization. The correspondences are limited inside corresponding regions, while still have similar vector field values. Note that some regions, e.g. hand palm, are not completely occupied. It is just because of different proportions. However, from the distribution of corresponding voxels, it can be expected that our correspondences can generate a reasonable warping function for morphing.

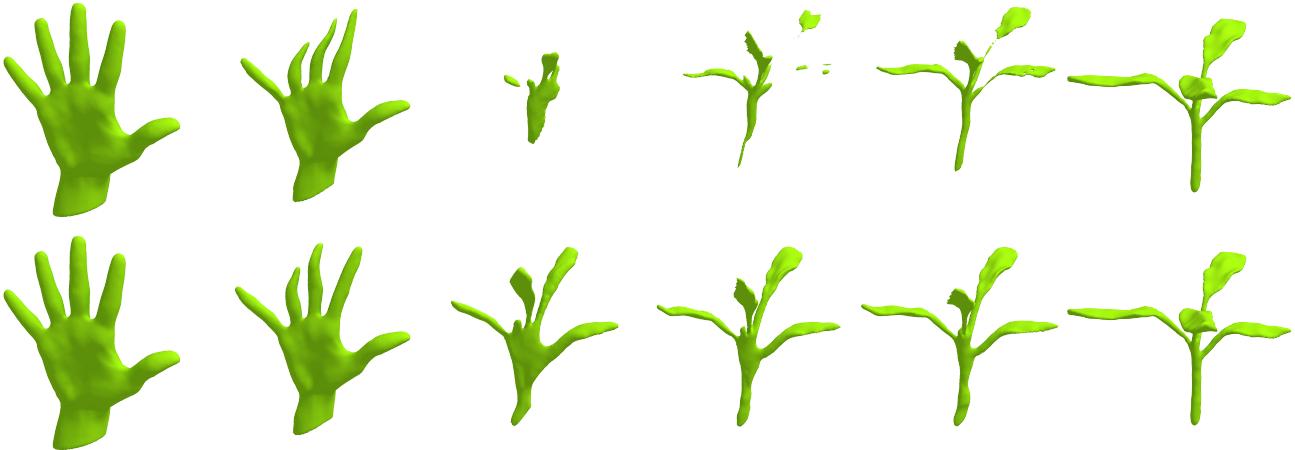


Figure 9: Morphing hand to plant. Top row: morphing sequence generated using traditional EMD optimization. The correspondence drift problem affects much on the results due to large proportion difference. Bottom row: morphing sequence produced based on our part-based EMD optimization algorithm.

#### 4.4. Skeleton-guided morphing

Now the dense correspondences in voxel level are established between source and target domains. As introduced in section 3, we can apply any available interpolation scheme to produce in-betweens. Since the ARAP DFI method [32] has many desired properties, it is exploited to generate the morphing sequences. The comparison of morphing sequences generated based on traditional EMD optimization and our method is illustrated in Figure 9. The results manifest that our part-based EMD optimization can generate better correspondences for warping functions.

## 5. Results and discussion

We implement our automatic part aware morphing algorithm using C++ on a desktop with Intel(R) Core(TM) i7-2700K CPU and 16 GB memory. The skeleton correspondence searching algorithm is implemented in Python. Volumetric data structure and tools in OpenVDB library [25] are employed to voxelize surface mesh and extract isosurface from level sets grid. The network simplex algorithm in [13] is exploited to optimize EMD problem, which essentially solves the minimum cost flow of a bipartite graph. Our algorithm is tested on several shape pairs with large geometry and structure variations. See the accompanying video for morphing sequences.

**Evaluation.** The first advantage of our method is that it avoids manually placing anchors on both source and target objects, which is tedious and time-consuming for users. It usually takes more than five minutes to place corresponding anchors and cannot guarantee to achieve good morphing sequence. Our algorithm finds anchor points automatically from matched skeleton feature nodes. But if skeletons have

different genus, our fully automatic method may get unsatisfactory results. To solve such problem, simple user interaction is needed to set up correspondences for branches forming cycles. It is still much easier than placing anchors manually without any hint. The last two examples in Figure 14 show morphing between objects with different genus.

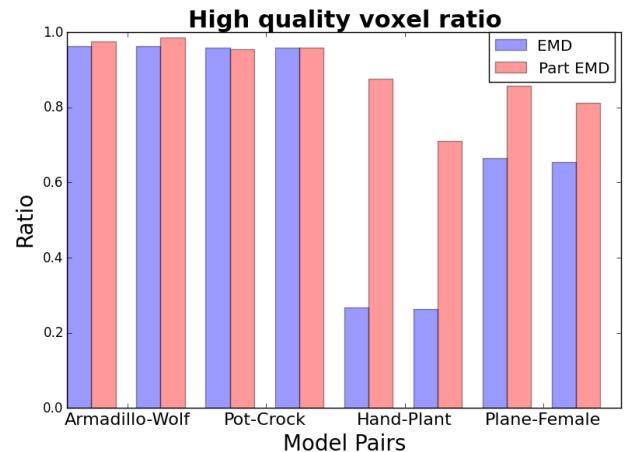


Figure 10: The ratios of voxels whose harmonic vector is close to their correspondences'. Frobenius norm is used to measure how close they are.

The second advantage is that our method can obtain high quality dense correspondences, even for objects have different proportions of corresponding parts. We compare the quality of correspondences generated by our part-based EMD optimization with traditional EMD optimization via the difference of harmonic vectors. In specific, for one

Model	EMD grid	#Int. voxel	#Anc	Time(mins)		
				EMD	FW	BW
Gecko	$80^3$	4954	30	0.37	0.08	2.41
Seal	$60^3$	6479				
Hand	$60^3$	8393	50	0.27	0.16	6.22
Plant	$125^3$	8944				
Dino.	$80^3$	8996	60	1.0	0.20	6.97
Horse	$60^3$	9936				
Lamp	$80^3$	9202	40	0.67	0.17	5.94
Genie	$60^3$	7445				
CAD	$30^3$	6435	30	0.62	0.09	2.46
Teapot	$40^3$	6910				
Arma.	$50^3$	8604	90	0.59	0.12	5.52
Wolf	$60^3$	6423				
Pot	$50^3$	9727	20	2.05	0.18	3.59
Crock	$50^3$	9062				
Plane	$75^3$	5757	40	0.23	0.06	2.29
Female	$65^3$	6712				
Vase1	$30^3$	7192	20	0.96	0.12	2.97
Vase2	$50^3$	8500				

Table 1: Various statistics in our experiments. EMD grid denotes the grid resolution used for EMD optimization. #Int. voxel is the number of interior voxels of the object. #Anc indicates the number of anchor points. The last three columns are time performance (in minutes) of EMD optimization, forward warping (FW) and backward warping (BW). EMD optimization needs to be computed once, but FW and BW should be computed for each frame. All results in this paper use the same resolution grid ( $150^3$ ) for DFI.

voxel and its correspondence, we compute the Frobenius norm of their harmonic vector difference and compare the value with a threshold ( $0.002 * \#Anchor$  in our experiments) to tell if the correspondence is good or not. The ratios of voxels that pass the test using EMD and our part-based EMD optimization are shown in Figure 10. It can be observed that our part-based EMD is generally better than traditional EMD, especially when corresponding parts have quite different proportions.

The cost of part-based EMD optimization is that EMD optimization should be done for each pair of parts. However, according to our experiments, EMD optimization takes relatively few time and just needs to be computed once. Table 1 shows several statistics of our experiments.

**Comparison.** We first compare our morphing result with a recent DFI framework [32]. Our algorithm finds anchor points automatically from skeletons, while their method requires users to manually place anchor points. Besides, our part-based EMD optimization can produce better morphing results, even if corresponding parts have highly dif-

ferent proportions. Figure 9 shows such case. The palm takes a large proportion of the hand, while the stem takes only a small proportion of the plant. Since anchor points on these two parts are corresponded to each other, EMD optimization may output dense correspondences with large drifts. This will produce undesired morphing results. Figure 1 shows morphing a gecko into a seal, whose structures are different. Figure 11 shows the morphing of two similar structure objects. However, geometric variation is very large and proportions of semantically corresponding parts are different. To make the comparisons objective, the same anchor points generated from our algorithm are provided for both algorithms.

We also made another comparison with morphing method proposed by Zhou *et al.* [38]. They realize shape blending by interpolating the profiles (*i.e.* generalized cylinders) which are associated with skeleton. In their method, they can only morph each pair of corresponding profile individually, the neighbouring consistency is not considered. Therefore, their morphing results may be not smooth in some regions (see rectangle regions in Figure 12). Figure 12 illustrates the comparison between their morphing results and ours. We test more data on our algorithm, and more results with large geometric variation and topology difference are shown in Figure 14.

**Limitation.** Although our method can automatically generate smooth and plausible morphing results even if the proportions of corresponding parts are quite different, it indeed needs meaningful anchor points to produce satisfying results. It means that our algorithm relies on the match of shape skeletons. If most feature nodes of skeletons are mismatched, the morphing sequence could be unpredictable.

Our strategy about automatically adding more anchor points works well on simplex skeleton (*i.e.* no cycles in skeleton). More anchors are only added on matched skeleton branches. But when it comes to complex skeleton with many cycles, there may exist several paths between two feature nodes making path correspondence ambiguous. Our algorithm fail to match these branches. However, with few user interaction (choose branch pairs), our framework can still produce good morphing results (last two examples in Figure 14). But if the numbers of skeleton branches are quite different, only selecting branch correspondences is not enough to produce satisfactory in-betweens due to the lack of anchors (Figure 13).

## 6. Conclusion

We have presented an automatic DFI morphing approach that smoothly blend source shape into target shape without the need of manually placing anchor points. The skeletons are first extracted from shapes, and then feature node correspondences are built. Based on these correspondences, more anchor points are added via parametrization. Thus,

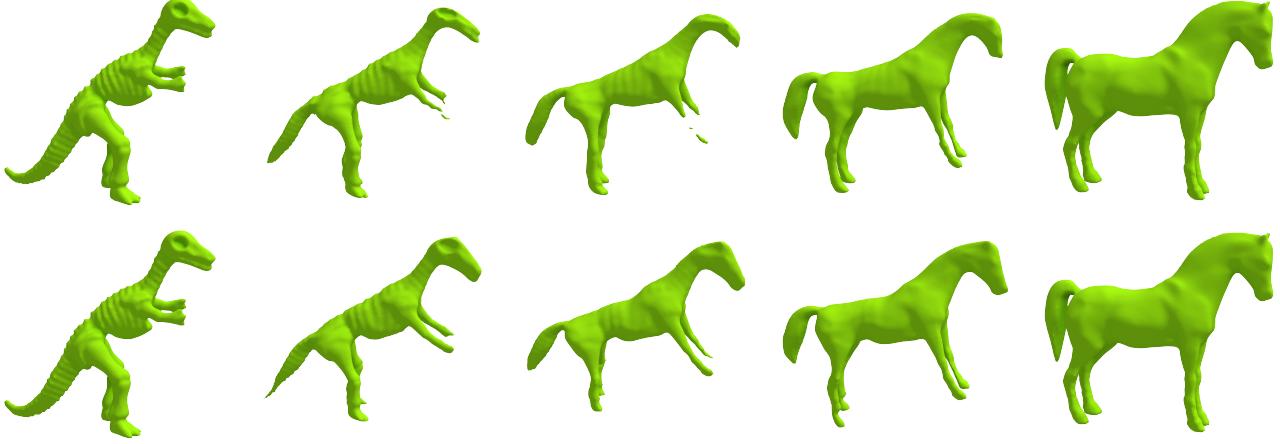


Figure 11: Comparison with Weng *et al.* [32]. Top row: morphing sequence from [32]. Bottom row: our morphing results.

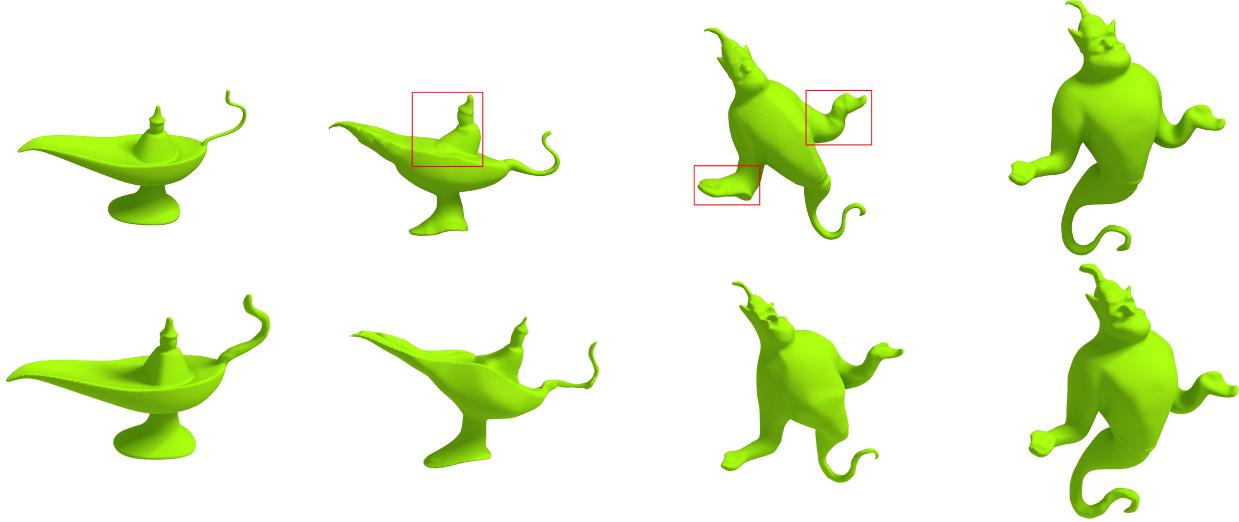


Figure 12: Comparison with Zhou *et al.* [38]. Top row: results from [38]. Bottom row: our results.

dense correspondences in voxel level can be established based on these meaningful anchors via EMD optimization. To alleviate large correspondence drifts, a part-based EMD optimization is proposed to generate more reasonable correspondences, which plays a key role in producing satisfactory morph sequences. Even if in some case, anchor points generated from our algorithm cannot produce perfect results, it can also provide users with feelings where to add more anchors to get better in-betweens.

Our method represents a step towards making 3D shape morphing process automatically and also with high quality. However, there are still much room for improvement. Although our method focuses on watertight models, it could easily extended to models with isolated parts. Besides, the key for satisfactory morphing is to establish reasonable correspondences. Therefore, in future work, we would like to

find a new way to efficiently build more accurate dense correspondences.

### Acknowledgement

We thank all the reviewers for their comments and feedback. We would also like to acknowledge our research grants: NSFC (61572507, 61202333, 61379103), 973 Program (2014CB360503), Guangdong Science and Technology Program (2015A030312015, 2014B050502009, 2014TX01X033), Hunan Innovation Program (CX2012B027), Shenzhen VisuCA Key Lab (CXB201104220029A).

### References

- [1] M. Alexa, D. Cohen-Or, and D. Levin. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th Annual Con-*

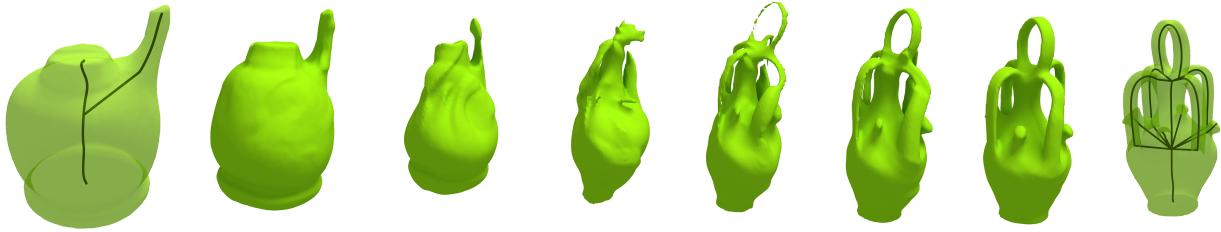


Figure 13: Our method produces unsatisfactory in-betweens due to the lack of anchors.

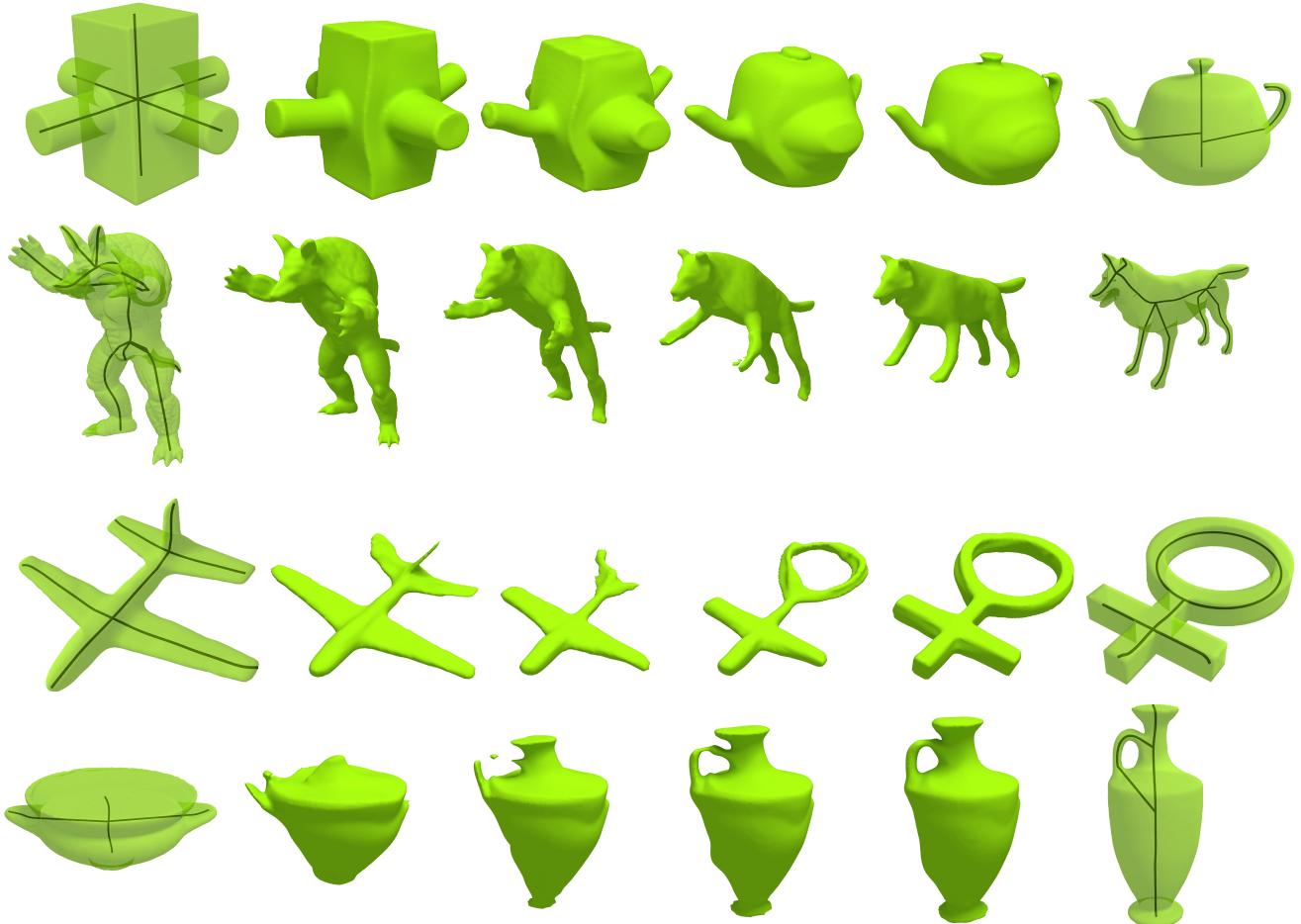


Figure 14: More morphing results generated by our method. Objects in top two rows have same genus 0 and can be morphed to their opponents automatically. The last two rows show morphing results with different genus, but correspondences of some skeleton branches need to be set up by user.

ference on Computer Graphics and Interactive Techniques, SIGGRAPH '00, pages 157–164, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 1, 2, 3

- [2] I. Alhashim, H. Li, K. Xu, J. Cao, R. Ma, and H. Zhang. Topology-varying 3d shape creation via structural blending. *ACM Trans. Graph.*, 33(4):158:1–158:10, July 2014. 2, 3
- [3] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: shape completion and animation of

people. *ACM Trans. Graph.*, 24(3):408–416, 2005. 2

- [4] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee. Skeleton extraction by mesh contraction. *ACM Trans. Graph.*, 27(3), 2008. 3
- [5] O. K.-C. Au, C.-L. Tai, D. Cohen-Or, Y. Zheng, and H. Fu. Electors voting for fast automatic shape correspondence. In *Computer Graphics Forum (In Proc. of Eurographics 2010)*, volume 29, pages 645–654, 2010. 2, 3, 4, 5

- [6] X. Bai and L. J. Latecki. Path similarity skeleton graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1282–1292, 2008. 3
- [7] T. Beier and S. Neely. Feature-based image metamorphosis. *SIGGRAPH Comput. Graph.*, 26(2):35–42, July 1992. 2
- [8] R. L. Blanding, G. M. Turkiiyah, D. W. Storti, and M. A. Ganter. Skeleton-based three-dimensional geometric morphing. *Comput. Geom. Theory Appl.*, 15(1-3):129–148, Feb. 2000. 3
- [9] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’99*, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 2
- [10] M. D. Buhmann and M. D. Buhmann. *Radial Basis Functions*. Cambridge University Press, New York, NY, USA, 2003. 4
- [11] D. Cohen-Or, A. Solomovic, and D. Levin. Three-dimensional distance field metamorphosis. *ACM Trans. Graph.*, 17(2):116–141, Apr. 1998. 1, 2, 3, 4
- [12] S. Coros, S. Martin, B. Thomaszewski, C. Schumacher, R. Sumner, and M. Gross. Deformable objects alive! *ACM Trans. Graph.*, 31(4):69:1–69:9, July 2012. 3
- [13] B. Dezs, A. Jüttner, and P. Kovács. Lemon - an open source c++ graph template library. *Electron. Notes Theor. Comput. Sci.*, 264(5):23–45, July 2011. 6, 7
- [14] E. Galin, A. Leclercq, and S. Akkouche. Morphing the blob-tree. *Comput. Graph. Forum*, 19(4):257–270, 2000. 1
- [15] L. Gao, Y. Lai, Q. Huang, and S. Hu. A data-driven approach to realistic shape morphing. *Comput. Graph. Forum*, 32(2):449–457, 2013. 1, 3
- [16] S.-M. Hu, C.-F. Li, and H. Zhang. Actual morphing: A physics-based approach to blending. In *Proceedings of the Ninth ACM Symposium on Solid Modeling and Applications, SM ’04*, pages 309–314, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association. 3
- [17] J. Huang, Y. Tong, K. Zhou, H. Bao, and M. Desbrun. Interactive shape interpolation through controllable dynamic deformation. *IEEE Trans. Vis. Comput. Graph.*, 17(7):983–992, 2011. 3
- [18] W. Jiang, K. Xu, Z. Cheng, and H. Zhang. Skeleton-based intrinsic symmetry detection on point clouds. *Graphical Models*, 75(4):177–188, 2013. 3
- [19] L. Kantorovich. On a problem of monge. *Journal of Mathematical Sciences*, 133(1):1383–1383, 2006. 2
- [20] D. Kravtsov, O. Fryazinov, V. Adzhiev, A. A. Pasko, and P. Comninou. Controlled metamorphosis between skeleton-driven animated polyhedral meshes of arbitrary topologies. *Comput. Graph. Forum*, 33(1):64–72, 2014. 1, 2
- [21] A. W. F. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution mesh morphing. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’99*, pages 343–350, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 2
- [22] A. Lerios, C. D. Garfinkle, and M. Levoy. Feature-based volume metamorphosis. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1995, Los Angeles, CA, USA, August 6–11, 1995*, pages 449–456, 1995. 1, 2
- [23] Z. Lian and J. Xiao. Automatic shape morphing for chinese characters. In *SIGGRAPH Asia 2012 Technical Briefs, SA ’12*, pages 2:1–2:4, New York, NY, USA, 2012. ACM. 3
- [24] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, Aug. 1987. 4
- [25] K. Museth. Vdb: High-resolution sparse volumes with dynamic topology. *ACM Trans. Graph.*, 32(3):27:1–27:22, July 2013. 7
- [26] G. Pasko, A. A. Pasko, and T. L. Kunii. Space-time blending. *Journal of Visualization and Computer Animation*, 15(2):109–121, 2004. 1
- [27] M. Sanchez, O. Fryazinov, T. Vilbrandt, and A. A. Pasko. Morphological shape generation through user-controlled group metamorphosis. *Computers & Graphics*, 37(6):620–627, 2013. 1, 2
- [28] C. Song, H. Zhang, X. Wang, J. Han, and H. Wang. Fast corotational simulation for example-driven deformation. *Computers & Graphics*, 40:49–57, 2014. 3
- [29] A. Tagliasacchi, I. Alhashim, M. Olson, and H. Zhang. Mean curvature skeletons. *Comp. Graph. Forum*, 31(5):1735–1744, Aug. 2012. 3
- [30] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011. 3
- [31] C. Von-Tycowicz, C. Schulz, H.-P. Seidel, and K. Hildebrandt. Real-time nonlinear shape interpolation. *ACM Trans. Graph.*, 34(3):34:1–34:10, May 2015. 1, 3
- [32] Y. Weng, M. Chai, W. Xu, Y. Tong, and K. Zhou. As-rigid-as-possible distance field metamorphosis. In *Computer Graphics Forum*, volume 32, pages 381–389. Wiley Online Library, 2013. 1, 2, 4, 5, 6, 7, 8, 9
- [33] D. Xu, H. Zhang, Q. Wang, and H. Bao. Poisson shape interpolation. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling, SPM ’05*, pages 267–274, New York, NY, USA, 2005. ACM. 1, 2, 3
- [34] K. Xu, H. Zhang, D. Cohen-Or, and Y. Xiong. Dynamic harmonic fields for surface processing. *Comput. Graph.*, 33(3):391–398, June 2009. 5
- [35] R. Zayer, C. Rössl, Z. Karni, and H. Seidel. Harmonic guidance for surface deformation. *Comput. Graph. Forum*, 24(3):601–609, 2005. 5
- [36] H. Zhang, A. Sheffer, D. Cohen-Or, Q. Zhou, O. van Kaick, and A. Tagliasacchi. Deformation-driven shape correspondence. In *Proceedings of the Symposium on Geometry Processing, SGP ’08*, pages 1431–1439, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association. 3, 4
- [37] Q. Zheng, A. Sharf, A. Tagliasacchi, B. Chen, H. Zhang, A. Sheffer, and D. Cohen-Or. Consensus skeleton for non-rigid space-time registration. *Computer Graphics Forum (Special Issue of Eurographics)*, 29(2):635–644, 2010. 3
- [38] Y. Zhou, K. Yin, H. Huang, H. Zhang, M. Gong, and D. Cohen-Or. Generalized cylinder decomposition. *ACM Trans. Graph.*, 34(6):171:1–171:14, Oct. 2015. 2, 3, 4, 5, 8, 9